

# Device App Access Points

## Endpoints Summary

Method	Path	Swagger
GET	<a href="#">/device_app_access_points/</a>	<a href="#">Swagger ↗</a>

“ The Device App Access Points API provides endpoints for retrieving and managing access point locations within your organization's network infrastructure. This API enables you to query access points with filtering capabilities, manage location-based services, and integrate physical access point data into your applications for monitoring, reporting, and operational management.

**Base URL:** <https://gate.zequenze.com/api/v1>

**Authentication:** All endpoints require a Bearer token:

```
Authorization: Bearer <your-api-token>
```

## Overview

The Device App Access Points API category focuses on managing physical access points within your organization's infrastructure. Access points represent physical locations where users can connect to services, authenticate, or access resources - such as Wi-Fi access points, entry gates, kiosks, or service terminals.

This API enables you to:

- Retrieve comprehensive lists of access points with detailed location and configuration data
- Filter access points by modification dates to sync changes
- Access geographic and organizational information for each access point

- Integrate access point data into monitoring dashboards, mobile applications, or reporting systems

Access points contain rich metadata including geographic coordinates, organizational details, physical addresses, and operational status. This makes the API particularly valuable for facility management, network monitoring, user experience optimization, and compliance reporting scenarios.

The pagination system allows efficient handling of large access point inventories, while filtering capabilities enable incremental synchronization and targeted queries based on your specific operational requirements.

---

# Endpoints

## GET /device\_app\_access\_points/

**Description:** Retrieves a paginated list of access points within your organization. This endpoint returns comprehensive information about each access point including location data, configuration details, and operational status. Use this endpoint to build access point inventories, create location-based services, or synchronize access point data with external systems.

**Use Cases:**

- Building a real-time dashboard showing all active access points across facilities
- Synchronizing access point data with external monitoring or management systems
- Creating mobile applications that help users locate nearby access points
- Generating compliance reports that include physical location and access point details
- Implementing location-based user authentication or service routing

**Full URL Example:**

```
https://gate.zequence.com/api/v1/device_app_access_points/?limit=25&offset=0&last_change__gte=2024-01-01T00:00:00Z
```

**Parameters:**

Parameter	Type	In	Required	Description
-----------	------	----	----------	-------------

last_change_gte	string	query	No	Filter access points modified on or after this date/time (ISO 8601 format). Useful for incremental synchronization.
limit	integer	query	No	Number of results to return per page. Default and maximum values depend on server configuration.
offset	integer	query	No	The initial index from which to return results. Use with limit for pagination.

### cURL Example:

```
curl -X GET "https://gate.zequenze.com/api/v1/device_app_access_points/?limit=25&offset=0" \
-H "Authorization: Bearer YOUR_API_TOKEN" \
-H "Content-Type: application/json"
```

### Example Response:

```
{
  "count": 156,
  "next": "https://gate.zequenze.com/api/v1/device_app_access_points/?limit=25&offset=25",
  "previous": null,
  "results": [
    {
      "id": 1001,
      "name": "Main Lobby WiFi Access Point",
      "short_name": "main-lobby-ap",
      "is_active": true,
      "description": "Primary wireless access point serving the main lobby and reception area",
      "organization_id": "org_12345",
      "thumbnail": "https://gate.zequenze.com/media/thumbnails/access_points/ap_1001.jpg",
      "address": "123 Corporate Boulevard",
      "reference": "Building A - Level 1 - Sector 3",
      "city": "San Francisco",
      "postal_code": "94105",
      "state": "California",
    }
  ]
}
```

```

    "region": "West Coast",
    "country_code": "US",
    "latitude": "37.7749",
    "longitude": "-122.4194",
    "created": "2024-01-15T09:30:00Z",
    "last_change": "2024-02-10T14:22:30Z"
  },
  {
    "id": 1002,
    "name": "Conference Room B Access Terminal",
    "short_name": "conf-room-b-terminal",
    "is_active": true,
    "description": "Secure access terminal for Conference Room B with integrated authentication",
    "organization_id": "org_12345",
    "thumbnail": "https://gate.zequenze.com/media/thumbnails/access_points/ap_1002.jpg",
    "address": "123 Corporate Boulevard",
    "reference": "Building A - Level 2 - Room B201",
    "city": "San Francisco",
    "postal_code": "94105",
    "state": "California",
    "region": "West Coast",
    "country_code": "US",
    "latitude": "37.7751",
    "longitude": "-122.4192",
    "created": "2024-01-20T11:45:00Z",
    "last_change": "2024-02-08T16:15:45Z"
  }
]
}

```

## Response Codes:

Status	Description
200	Success - Returns paginated access points data
401	Unauthorized - Invalid or missing Bearer token
403	Forbidden - Token valid but insufficient permissions for this organization
404	Not Found - Endpoint not available

Status	Description
422	Unprocessable Entity - Invalid query parameters (e.g., malformed date format)

# Common Use Cases

## Use Case 1: Real-Time Access Point Monitoring Dashboard

Build a monitoring dashboard that displays all active access points with their current status, location, and last update times. Use pagination to efficiently load large numbers of access points and implement real-time updates by periodically querying with `last_change__gte` to fetch only recently modified access points.

## Use Case 2: Mobile App Location Services

Develop a mobile application that helps users find nearby access points. Retrieve all access points with geographic coordinates, then use the latitude/longitude data to calculate distances and display the closest available access points to the user's current location.

## Use Case 3: Incremental Data Synchronization

Synchronize access point data with external systems like network monitoring tools or facility management software. Use the `last_change__gte` parameter to fetch only access points that have been modified since your last synchronization, reducing bandwidth and processing overhead.

## Use Case 4: Compliance and Audit Reporting

Generate compliance reports that require detailed location and access point information. Retrieve comprehensive access point data including addresses, references, and organizational details to create audit trails and compliance documentation.

## Use Case 5: Geographic Access Point Distribution Analysis

Analyze the geographic distribution of access points across different regions, cities, or facilities. Use the location data (city, state, region, country\_code) to group access points and generate

# Best Practices

- **Implement Pagination Efficiently:** Use reasonable limit values (25-100 items per page) to balance performance and user experience. Always check for the `next` field to determine if additional pages are available.
- **Use Incremental Synchronization:** When building systems that need to stay synchronized with access point data, use the `last_change__gte` parameter to fetch only recently modified records, reducing API calls and improving performance.
- **Handle Geographic Data Appropriately:** Latitude and longitude values are returned as strings in decimal format. Convert to appropriate numeric types in your application and validate coordinates before using them in mapping or distance calculations.
- **Cache Access Point Data:** Access point configurations typically don't change frequently. Implement appropriate caching strategies with cache invalidation based on `last_change` timestamps to reduce unnecessary API calls.
- **Monitor Rate Limits:** Implement proper error handling for rate limiting scenarios and consider implementing exponential backoff strategies for retrying failed requests.
- **Validate Country Codes:** The API supports a comprehensive list of ISO country codes including special territories and regions. Ensure your application can handle the full range of possible `country_code` values when implementing location-based features.

---

Revision #4

Created 2026-02-04 05:13:00 UTC by ipena@zequenze.com

Updated 2026-02-11 03:13:01 UTC by ipena@zequenze.com